DOCUMENT RESUME

ED 272 186 IR 012 312

AUTHOR Gray, Peter J.
TITLE Microcomputers: Instrument Generation Software. Evaluation Guides. Guide Number 11.
INSTITUTION Northwest Regional Educational Lab., Portland, Oreg.
SPONS AGENCY National Inst. of Education (ED), Washington, DC.
PUB DATE [84]
CONTRACT 400-80-0105.
NOTE 9p.; A product of the Research on Evaluation Program.
PUB TYPE Guides - Non-Classroom Use (055)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Software; *Data Collection; *Evaluation Criteria; Evaluation Methods; Information Needs; *Information Retrieval; Information Utilization; *Microcomputers; Recordkeeping
IDENTIFIERS *Software Evaluation

ABSTRACT

Designed to assist evaluators in selecting the appropriate software for the generation of various data collection instruments, this guide discusses such key program characteristics as text entry, item storage and manipulation, item retrieval, and printing. Some characteristics of a good instrument generation program are discussed; these include easy text entry and editing for item creation and modification, convenient and safe data storage, simple item retrieval, and flexible printing options. In addition, four types of item modification are identified: insertion or deletion of material, reformatting, adding options, and changing the item's descriptors. Four instrument generation programs are briefly described, including the producer, cost, and hardware required. Suggestions are provided for selecting the right software, including comments on hardware requirements, operating systems, versatility, and price. It is noted that selection should be based on a consideration of the combination of features most desired for the types of tasks to be performed using the program. A list of seven suggestions for making a sound choice of software and a five-item bibliography complete the document. (JB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED272186

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy

EVALUATION GUIDES

Guide Number 11

MICROCOMPUTERS: INSTRUMENT GENERATION SOFTWARE

Peter J. Gray

The use of microcomputer based instrument development programs is discussed, including:

- Key Characteristics of Instrument Generating Software
- Instrument Generation Sampler
- Selecting the Right Software

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Jerry D. Kirkpatrick

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Research on Evaluation Program Nick L. Smith, Director

Northwest Regional Educational Laboratory
300 S.W. Sixth Avenue • Portland, Oregon 97204 • Telephone (503) 248-6800

IR042312

KEY CHARACTERISTICS OF INSTRUMENT GENERATING SOFTWARE

Generic software such as word processing, data base management, spreadsheet, and telecommunications have many uses in evaluation research. There are also some very specific programs that are especially relevant to evaluation efforts. Of special usefulness are programs designed to assist in the development of various data collection instruments. A good instrument generation program should have the following characteristics:

- easy text entry and editing for item creation and modification;
- convenient and safe data storage so that items can be quickly manipulated and reliably saved;
- simple item retrieval to facilitate the development of a complete instrument;
- flexible printing options for generating instruments, including last-minute editing of items, as well as the creation of headings and special directions;

Text Entry

Flexibility in text entry is the key characteristic that a program needs in order to ease the task of developing items. A good program will allow both immediate modification of items and modification subsequent to their creation. Types of modification might include:

- inserting or deleting material,
- reformatting,
- adding options (i.e., responses),
- changing the item's descriptors,

These characteristics facilitate the continuing improvement of items throughout the life of an item bank.

Another important aspect of flexible text entry is the variety of item formats that are allowed. For example, programs differ in their ability to handle true/false, multiple choice, matching, fill-in-the-blank, or essay items.

In addition to item type, the provisions made by a program for the inclusion of pictorial material is one that must be considered if such material is critical to the nature of an

item. Pictures, diagrams, formulas with subscripts and superscripts, and special symbols are almost impossible to save in machine-readable format. About the only ways to overcome this limitation are (1) for a program to allow one to add a reminder to an item that pictorial material is to be included at the time of printing, and (2) for the program to allow designation of the number of line feeds to follow an item so that room is provided for the material.

Item Storage and Manipulation

Safe item storage is related to a program's ability to verify new material before it is written to disk and to check before items are saved to insure that another set with the same name is not inadvertently destroyed.

There are two ways to handle the storage and manipulation of test items. One way is to allow the creation of a large number of items, limited only by the capacity of the auxiliary storage medium, such as a floppy disk drive. In this case, items are manipulated by transferring as many as possible to random access memory (RAM) and gradually working through the item bank. The drawback here is the time needed to transfer items back and forth from auxiliary storage to RAM.

The other way to store and manipulate items is to limit each file to as many items as will fit in RAM at one time. This allows faster manipulation of a given set, but, of course, the set is smaller than a file whose size is related to disk capacity. And, if items from separate sets scattered throughout a disk are to be manipulated, then the task can be even more time consuming than a similar task performed with a program which is based around disk storage limits.

Some programs that are RAM based provide for the merging of subfiles into larger units, and for breaking bigger files into smaller ones. This can facilitate the task of manipulating a given set of items.

Item Retrieval

The simplest method of retrieving items is to assign each one a number as it is created. Then, using a master printout, one can locate and call up each item by its number in response to a prompt provided by the program. Another way is to be able to display items on the screen in numeric order and to be able to press a single key to identify an item to be selected. For large item banks, the use of item descriptors can help to narrow down the number of items to be reviewed in order to identify those to

be selected. For example, descriptors might refer to (1) type of item (i.e., item format), (2) subject matter, or (3) classification in relation to a taxonomy (e.g., Bloom's taxonomy).

In some programs, the answer line can be used not only for storing the specific answer to an item, but also for information that can be used to help find items. For example, the following might be stored in the answer line:

> key words in an essay, notes to oneself or any alphanumeric information, or if the items are based on a set of objectives, the abbreviated name or number of the objective tested by the item.

Another piece of information that would be useful in retrieving appropriate items for a given instrument is an indication of item difficulty. This could be accomplished by simply placing in the answer line a rating of "easy," moderate," or "hard." A more sophisticated method would be the ability to include both difficulty and discrimination indexes for all item alternatives. Using a sort routine, items could then be selected on index values, as well as information on content, format, and other characteristics.

This inclusion of specific information useful for item retrieval adds a whole new dimension to instrument-generation programs.

Printing

Once items are retrieved from the bank for a particular application, it is helpful if they can be edited to tailor them to that setting. It would also be useful to have the option of saving the edited items in a separate file for later use. In any case, the original set of items in the item bank itself should not be permanently altered by these last-minute modifications. This is especially important if the item bank is intended to be generic; for example, if it is a set of essay items on current events where the details of the items are to be added in relation to the latest happenings.

In addition to editing capabilities, a useful program will allow one to format an instrument to a particular situation. Using special headings or titles is one way to accomplish this. Being able to tailor the instructions can greatly improve the utility of an item bank, too. The capability to determine left and right margins, to determine the number of lines of print and total number of lines per page, to require a pause after each page is printed, and to print all or just part of a test are all program print features that can help improve the face validity of an instrument.

Summary

Programs exhibit the above characteristics to varying degrees. Trying different programs is the only way to determine whether the features you need are included and readily usable.

INSTRUMENT GENERATION SAMPLER

Program/Company	Cost	System	Comments
Author I Radio Shack Education Division	$150	TRS-80 I/III	Test assembly, on-line testing, and the development of student records/profiles are features of this program
The Learning System, Microlab Highland Park, Illinois	$150	Apple	On-line testing of items from a program item bank, class statistics and student profiles are all part of this program.
Test Bank 2.1 Advanced Technology Applications San Diego, CA	$450	TRS-80 I/III	Flexible text entry, item editing, test preparation are just some of the features of this program
Teacher Utilities Vol. 1 Minnesota Educational Computing Consortium (MECC)	$37	Apple II	Item files, on-line testing, test assembly, test printing, plus class statistics and grades are features of this program.

SELECTING THE RIGHT SOFTWARE

In one sense, hardware requirements are the program characteristic to consider first, and in another sense, they are the characteristic to consider last. From a realistic point of view, the first criteria for selecting a program is whether it will run on a machine you already have, or on a machine that you expect to buy for statistical and other purposes. However, within these general constraints, hardware becomes a secondary consideration, because there is a variety of good programs to choose from for most of the popular and widely used machines with operating systems such as Apple DOS 3.3, IBM-PC DOS or MS-DOS, and CP/M. Within each group, programs vary in terms of their sophistication and cost and in terms of the specific hardware system characteristics that they require.

In summary, if a commitment has already been made to purchase a particular machine, or if there are special budget limitations, hardware-related requirements are the first features of a program that should be considered. However, if there are no rigid constraints, it is best to ignore these requirements for the time being and move on to the other more substantive features of instrument generation programs.

It is important to evaluate a program in terms of its versatility regarding those features you need most. Selection may come down to the program(s) with the best ratings on those features of greatest importance as opposed to those with the best over-all ratings. This notion of the highest ratings on the most important features is worth considering. Sometimes pricing, especially in regard to multiple copies, is the deciding factor among programs of generally equal ratings. In other cases it may be that speed, error handling, and versatility (i.e., program performance) is more important than either ease of use or support. Therefore, lower ratings in these areas would not disqualify a program if it was a strong performer.

Any combination of features is possible. Selection should be based, therefore, on a consideration of the combination of features most desired for the types of tasks to be performed using the program.

In order to make a sound choice:

1. Describe your use(s) - what will you use the program for?

2. Identify the features you need - what do you want to be able to do?

3. Plan ahead for new needs - what are you likely to want a year from now?

4. Consider constraints - What price range, hardware (e.g., machine type, printer features) and user preferences are you limited by?

5. Put features into a rough priority list - which are the most, somewhat, and least important features?

6. Try out and compare products - which ones have the features you need and want within your constraints?

7. Remember support - will there be someone you can talk to if there are problems after you buy the program?

REFERENCES

Deck, D. & Estes, G. D. (1984, April). Microcomputer software support for item banking. A paper presented at the annual meeting of the American Educational Research Association, New Orleans, LA. Portland, OR: Northwest Regional Educational Laboratory, Assessment and Evaluation Program.

Estes, G. D. & Arter, J. A. (1984, June). A guide to item banking in education (2nd ed.). Portland, OR: Northwest Regional Educational Laboratory, Assessment and Evaluation Program,

Estes, G. D. & Arter, J. A. (Nd.) Item banking for state and local test development and use: Specific guidelines for local practitioners. Portland, OR: Northwest Regional Educational Laboratory, Assessment and Evaluation Program.

Hsu, T., & Nitko, A. (1983, Winter). Microcomputer testing software teachers can use. Educational measurement: Issues and practices, 2(4), 15-18, 23-30.

Lippey, G. (Ed.) (1974). Computer-assisted test construction. Englewood Cliffs, NJ: Educational Technology Publications.

RECENT GUIDEBOOKS IN THIS SERIES

No.	Title
1	Microcomputers and Evaluation
2	Cost-Outcome Analysis: Measuring Costs
3	Microcomputers: Word Processing
4	Cost-Outcome Analysis: Measuring Outcomes
5	Microcomputers: Statistical Analysis Software
6	Investigative Journalism Techniques
7	Microcomputers: Data Base Management Software
8	Committee Hearings: Their Use in Evaluation
9	Microcomputers: Spreadsheet Software
10	Methods of Product Evaluation
11	Microcomputers: Instrument Generation Software
12	Evaluation Design: The Evaluator and the Architect
13	Microcomputers: Communication Software
14	Evaluation Design: Selecting Methods
15	Service Delivery Assessment Techniques

These materials are in the public domain and may be reproduced without permission. The following acknowledgment is requested on materials which are reproduced: Developed by the Northwest Regional Educational Laboratory, Portland, Oregon.

Printed by the Northwest Regional Educational Laboratory, a private nonprofit corporation. The work upon which this publication is based was performed pursuant to Contract No. 400-80-0105 of the National Institute of Education. It does not, however, necessarily reflect the views of that agency.

Research on Evaluation Program
Northwest Regional Educational Laboratory
300 S. W. Sixth Avenue
Portland, OR 97204